Configure Failure−Handling and Server−Unreachable Mechanisms for OCS Failure Resolution on the ASR5K

TAC **Document ID: 118993**

Contributed by Shashank Varshney, Cisco TAC Engineer.
Jun 02, 2015

Contents

Introduction
Prerequisites
Requirements
Components Used
Background Information
Configure
Network Diagram
Tx−Expiry
Response Timeout
Diameter Session Failover
FH Mechanism
FH Mechanism Configuration
FH Mechanism Default Behavior
FH Mechanism Detailed Call Flow
SU Mechanism
SU Mechanism Configuration
SU Mechanism Call Flows
FH and SU Example Configurations
Verify
Troubleshoot
Related Information

Introduction

This document describes how to configure the Failure−Handling (FH) and Server−Unreachable (SU) mechanisms on the Gy interface in order to resolve issues that are encountered on the Online Charging System (OCS) or in regards to connectivity between the Policy and Charging Enforcement Function (PCEF) and the OCS. The information that is described in this document is applicable to the Home Agent (HA), Gateway General Packet Radio Service (GPRS) Support Node (GGSN), and Packet Data Network Gateway (PGW) functionalities that run on the Cisco 5000 Series Aggregated Services Router (ASR5K).

Prerequisites

Requirements

Cisco recommends that your system meet these requirements in order to use the FH and SU mechanisms:

- The Enhanced Charging Service (ECS) is available

- The PCEF exists within the HA, GGSN, or PGW
- There is proper diameter connectivity via the diabase
- The Diameter Credit Control Application (DCCA) is available

Components Used

The information in this document is based on all versions of the ASR5K.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

Background Information

The PCEF is connected to the OCS over the Gy interface, which uses Diameter as the base protocol and DCCA. This is the message flow between the PCEF and the OCS:

- ***Credit Control Request (CCR)*** This message is sent from the PCEF to the OCS. There are three types of CCR messages: Initial, Update, and Terminate.
- ***Credit Control Answer (CCA)*** This message is sent from the OCS to the PCEF in response to the CCR. There are also three types of CCA messages: Initial, Update, and Terminate.
- ***Re−Authorization Request (RAR)*** This message is sent from the OCS to the PCEF when a session re−authorization is required.
- ***Re−Authorization Answer (RAA)*** This is the response to the RAR from the PCEF to the OCS.

Diameter connectivity is established between the PCEF and the OCS in order to enable the message flow. There is a possibility that the OCS might send negative messages, the transport connection might fail between the PCEF and the OCS, or the message might timeout, which can cause a failure in the subscriber session establishment. This can prevent the subscriber from the use of services.

These two mechanisms can be used in order to resolve this issue:

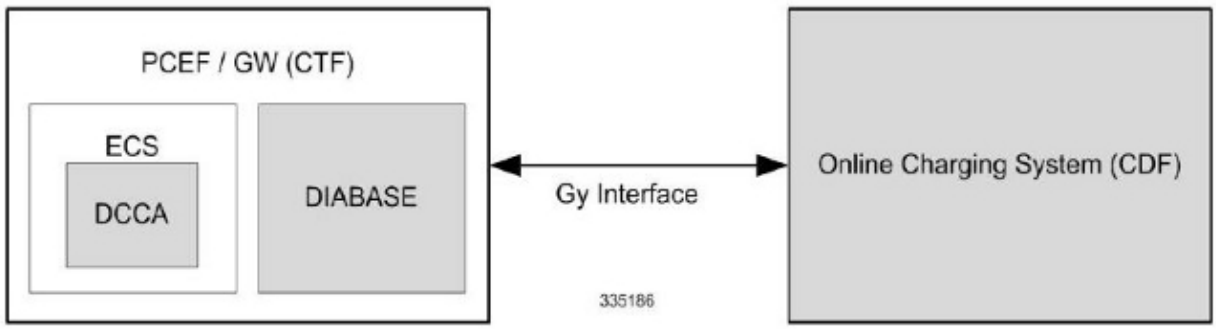
- The FH mechanism
- The SU mechanism

Configure

This section describes the configurations that are required in order to support the FH and SU mechanisms.

Network Diagram

The information in this document uses this topology:



Tx−Expiry

This is an application−level timer for the DCCA that is configurable in the *diameter credit−control* settings. The value can range between 1 and 300 seconds.

Here is an example:

```
[local]host_name(config−dcca)# diameter pending−timeout <value>
```

Response Timeout

This is a diabase timeout and is configurable in the *Diameter Endpoint* settings. The value can range between 1 and 300 seconds.

Note: The value that is configured for this timer should be greater than that used for the Tx−Expiry timer.

Here is an example:

```
[context_name]host_name(config−ctx−diameter)# response−timeout <value>
```

Diameter Session Failover

This feature is used in order to enable or disable the diameter credit control session failover, which allows the system to use a secondary server when the primary server becomes unreachable. This is configurable in the *diameter credit−control* settings.

Here is an example:

```
local]host_name(config−dcca)# diameter session failover
```

FH Mechanism

The FH mechanism only operates if diameter session failover is present. The FH allows the system to choose whether to continue the session and convert to offline, or to terminate the session when a connection or message−level error occurs.

Note: The FH is enabled and configured by default.

FH Mechanism Configuration

The FH mechanism can be configured in the *diameter credit−control* settings. Here is the syntax that is used in the FH configuration:

```
failure-handling { initial-request | terminate-request | update-request } { continue
 [ go-offline-after-tx-expiry | retry-after-tx-expiry ] | retry-and-terminate,
 [ retry-after-tx-expiry ] | terminate }
```

The first section specifies the *request type*: Initial (CCR−I), Update (CCR−U), and Terminate (CCR−T).

The second section specifies the *action* that should be performed when the FH mechanism is activated. These three actions are possible with the FH mechanism:

- ***Continue*** This allows the session to continue and converts it to offline. This function uses two options that are related to Tx−expiry:
  - ♦ ***Go−offline−after−tx−expiry*** This starts offline charging after the Tx−expiry occurs.
  - ♦ ***Retry−after−tx−expiry*** This retries the secondary server after the Tx−expiry occurs.
- ***Retry−and−terminate*** This terminates the session after the system retries the secondary server, if the secondary server is not available either. This also uses the ***Retry−after−tx−expiry*** option, which retries the secondary server after the Tx−expiry occurs.
- ***Terminate*** This terminates the session without any attempts to contact the secondary server.

FH Mechanism Default Behavior

This section describes the FH default behavior when no configuration is present. By default, the FH mechanism is activated during a Response Timeout (RT), except when the *Terminate* action is configured.

If a *Credit−Control−Failure−Handling* Attribute Value Pair (AVP) is received from the server, then the received settings are applied.

Here are some examples:

- ***Initial−Request > Terminate***
- ***Update−Request > Retry−and−Terminate***
- ***Terminate−Request > Retry−and−Terminate***

FH Mechanism Detailed Call Flow

This section describes the detailed call flow of the FH mechanism with all possible options.

Initial−Request

CCFH Setting	CLI Command	Behavior at Tx	Behavior at RT	Secondary is Up	Secondary is Down
Continue	initial−request continue	N/A	Continue	Secondary takes over after RT	Offline after another RT. No more quota requests are performed for any rating group within the session after DCCA failure (even if connectivity to DCCA is restored)

	initial−request continue go−offline−after−tx−expiry	Offline	N/A	Offline at Tx	Offline at Tx
	initial−request continue retry−after−tx−expiry	Continue	N/A	Secondary takes over after Tx	Offline after another Tx
	initial−request retry−and−terminate	N/A	Retry	Secondary takes over after RT	Terminate after another RT
Retry−and−terminate	initial−request retry−and−terminate retry−after−tx−expiry	Retry	N/A	Secondary takes over after Tx	Terminate after another Tx
Terminate	initial−request terminate	Terminate	N/A	Terminate after Tx	Terminate after Tx

Update−Request

CCFH Setting	CLI Command	Behavior at Tx	Behavior at RT	Secondary is Up	Secondary is Down
	update−request continue	N/A	Continue	Secondary takes over after RT	Offline after another RT
Continue	update−request continue go−offline−after−tx−expiry	Offline	N/A	Offline at Tx	Offline at Tx
	update−request continue retry−after−tx−expiry	Continue	N/A	Secondary takes over after Tx	Offline after another Tx
	update−request retry−and−terminate	N/A	Retry	Secondary takes over after RT	Sends CCR−T after another RT
Retry−and−terminate	update−request retry−and−terminate retry−after−tx−expiry	Retry	N/A	Secondary takes over after Tx	Sends CCR−T after another Tx
Terminate	update−request terminate	Terminate	N/A	Sends CCR−T after Tx	Sends CCR−T after Tx

Terminate−Request

CCFH Setting	CLI Command	Behavior at Tx	Behavior at RT	Secondary is Up	Secondary is Down

Continue	terminate−request continue	N/A	Retry	CCR−T is sent to secondary after RT	Terminate after another RT
	terminate−request continue go−offline−after−tx−expiry	Retry	N/A	CCR−T is sent to secondary after Tx	Terminate after another Tx
	terminate−request continue retry−after−tx−expiry	Retry	N/A	CCR−T is sent to secondary after Tx	Terminate after another Tx
Retry−and−terminate	terminate−request retry−and−terminate	N/A	Retry	CCR−T is sent to secondary after RT	Terminate after another RT
	terminate−request retry−and−terminate retry−after−tx−expiry	Retry	N/A	CCR−T is sent to secondary after Tx	Terminate after another Tx
Terminate	terminate−request terminate	Terminate	N/A	Terminate after Tx	Terminate after Tx

SU Mechanism

The SU mechanism is similar the FH mechanism, but it provides more granular control over failure procedures. In addition to the continuation of the session after the message− and connection−level (transport) failures, this mechanism can be used when the responses are slow from the OCS. It also provides the options to either continue the session for some time duration/quota exhaustion before termination, or use configurable interim quota (volume and time) and configurable server retries before a session is converted to offline or terminated.

SU Mechanism Configuration

The SU mechanism can be configured in the *diameter credit−control* settings. The syntax that is used in the SU configuration varies in accordance with the version that is used.

For Versions 12.1 and earlier, this is the syntax that is used for the SU mechanism configuration:

```
servers-unreachable { initial-request { continue | terminate [ after-timer-expiry
 <timeout_period> ] } | update-request { continue | terminate [ after-quota-expiry
 | aftertimer-expiry <timeout_period> ] } }
```

For Versions 12.2 and later, this is the syntax that is used for the SU mechanism configuration:

```
servers-unreachable { behavior-triggers { initial-request | update-request }
 result-code { any-error | <result-code> [ to <end-result-code> ] }
 | transport-failure [ response-timeout | tx-expiry ] | initial-request
 { continue [ { [ after-interim-time <timeout_period> ] [ after-interim-volume
 <quota_value> ] } server-retries <retry_count> ] | terminate [ {
 [ after-interim-time <timeout_period> ] [ after-interim-volume <quota_value> ]
 } server-retries <retry_count> | after-timer-expiry <timeout_period> ] }
 | update-request { continue [ { [ after-interim-time <timeout_period> ]
 [ after-interim-volume <quota_value> ] } server-retries <retry_count> ]
```

```
| terminate [ { [ after-interim-time <timeout_period> ] [ after-interim-volume
<quota_value> ] } server-retries <retry_count> ] | after-quota-expiry |
after-timer-expiry <timeout_period> ] } }
```

Note: In versions prior to Version 12.2, there was flexibility to configure the FH and SU mechanisms independently; however, in Versions 12.2 and later, the SU mechanism takes precedence over the FH mechanism when configured.

If the server returns the CC−FH AVP, and the SU mechanism is configured for a set of behavior triggers, then the SU configuration is applied; otherwise, the CC−FH AVP value is applied. By default, result codes such as 3002, 3004, and 3005 fall under *delivery failure* and are treated as RTs.

These actions are possible with the SU mechanism:

- ***Behavior−Trigger*** This specifies the type of messages that can be Initial−Requests (CCR−I) or Update−Requests (CCR−U). There are three options available for these triggers:
    - ♦ ***Result−Code*** This allows the configuration of specific result codes, range of result codes (3000−5999), or any error along with the message type.
    - ♦ ***Transport−Failure*** This specification triggers the behavior upon transport failure, which is backwards compatible with Version 12.0. There are two options available for this setting:
        - ◊ ***Response−Timeout*** This option triggers the behavior upon RT and should always be used with transport failures.
        - ◊ ***Tx−Expiry*** This option triggers the behavior upon Tx−expiry and should always be used with transport failure.
    - ♦ ***Actions*** This specifies the action that is performed when an SU trigger for CCR−I or CCR−U occurs. This action varies in accordance with the message type and software version.
- ***Continue*** This allows the session to be converted to offline and continue. There is no further options available for the use of this action in versions prior to Version 12.2. In Versions 12.2 and later, the interim quota, server−retries, and after−timer−expiry options are available for configuration with this action.
- ***Terminate*** This results in termination of the session when the server becomes unreachable. This action allows the interim quota, server−retries, and after−timer−expiry options.

These options can be used with the *Continue* or *Terminate* action:

- ♦ ***After−interim−time*** This option allows continuance or termination of the call after the interim timeout period. This is similar to a time quota before the action is performed. The value is formatted in seconds and can range between 1 and 4,294,967,295.
- ♦ ***After−interim−volume*** This option assigns the interim quota and allows the continuance or termination of the session before exhaustion of the configured volume. The value is formatted in bytes and can range between 1 and 4,294,967,295.
- ♦ ***Server−retries*** This option allows the PCEF to retry the OCS before continuance or termination of the session. The retry count can be configured, and the value ranges between 0 and 65,535. If the value is zero, then the retry does not occur and the action is immediately applied.

Note: The *after−interim−time* and *after−interim−volume* options are always used with the *server−retries* option, or all three can be used simultaneously and applied to both continue and terminate actions. The *after−interim−time* and *after−interim−volume* options also assign time as well as volume quota, and the quota that is exhausted first triggers the server retry.

- ***After−timer−expiry*** This option specifies the time duration (in seconds) for which sessions remain in the offline status before termination occurs. The values can range between 1 and 4,294,967,295. This option is only applicable for terminate actions.

- ***After−quota−expiry*** This option terminates the session upon exhaustion of the already assigned quota.

Here is some important information to remember:

- The *after−interim−time*, *after−interim−volume*, and *server−retries* options can be used individually or in combination, and they are applicable to both continue and terminate actions.

- The *after−quota−expiry* option is only applicable for the Update−Requests behavior trigger.

- The *after−timer−expiry* option is only applicable for the terminate action.

- The *after−interim−time*, *after−interim−volume*, and *server−retries* options are only applicable for Versions 12.2 and later.

- If diameter session failover is supported (and configured), then the secondary server is always contacted before the SU mechanism is triggered.

- The server that was contacted last before the SU mechanism is triggered is always contacted when the interim time or interim volume is exhausted and the *server retries* option is configured with a value that is greater than zero. For example, if OCS1 is tried first, and OCS2 is tried after an error at OCS1, then communication with OCS2 triggers the SU mechanism. During the server retry attempt, OCS2 is contacted first and then OCS1 if a negative response is received from OCS2.

SU Mechanism Call Flows

The SU mechanism can be triggered by a failure of the CCR−I or the CCR−U, and the cause can be an error code, transport failure, Tx−expiry, or RT. The action can be an allocation of interim quota (time and/or volume), server retries count, timer value (causes the session to go offline for specified time and only for termination), or quota expiry (only for the CCR−U and termination) before the session goes offline or terminates.

The interim quota is provided on a per−session basis, not a per−Rating Group (RG) basis in Multiple Services Credit Control (MSCC) scenarios.

There is a possibility that the primary server triggers transport failure and the secondary server triggers the Tx−expiry or response−timeout. In this scenario, the latest error is considered to be the trigger of the failure.

If the SU mechanism is not configured for any failure trigger, then the FH mechanism is triggered.

Note: The sections that follow provide some call flow examples that are related to the SU mechanism. These call flows are provided under the assumption that diameter−session−failover is supported and the secondary server is configured with a Tx−expiry value that is less than the RT value. Also, it is assumed that the SU mechanism is configured for transport−failure, Tx−expiry, and RT.

Initial−Request without Session Disconnect

Here is the message flow for this scenario:

1. The PCEF sends a CCR−I to the OCS.

2. A timeout or transport failure is detected. If transport failure is detected, then the PCEF immediately retries with the secondary server; otherwise, the Tx−expiry is triggered.

3. If the secondary server also has a transport failure or timeout, then the SU mechanism is triggered. This occurs immediately for transport failures, or after the Tx−expiry for a timeout.

4. If the interim volume and/or time are configured, then the interim quota is assigned to the session.

5. After exhaustion of the interim quota (time or volume) and if the *server retries* value is greater than zero, then the CCR−I is again sent to the server that triggered the SU mechanism. If there is another failure, the CCR−I is sent to another server.

6. If the transport failure or Tx−timeout is again detected, then Steps 2 through 5 are repeated until the *server retries* value is exhausted or a successful response does not come from the OCS.

7. If the issue still exists, then the session is continued (converted to offline) or terminated as per the configuration.

Note: The interim quota that is consumed while the session goes into SU mode due to CCR−I is not reported in the next CCR−I. All of the used interim quota is reported in the CCR−U, which follows the successful CCA−I.

Initial−Request with Session Disconnect

Here is the message flow for this scenario:

1. The PCEF sends a CCR−I to the OCS.

2. A timeout or transport failure is detected. If transport failure is detected, then the PCEF immediately retries with the secondary server; otherwise, the Tx−expiry is triggered.

3. If the secondary server also has a transport failure or timeout, then the SU mechanism is triggered. This occurs immediately for transport failures, or after the Tx−expiry for a timeout.

4. If the interim volume and/or time are configured, then the interim quota is assigned to the session.

5. After exhaustion of the interim quota (time or volume) and if the *server retries* value is greater than zero, then the CCR−I is again sent to the server that triggered the SU mechanism. If there is another failure, the CCR−I is sent to another server.

6. If the transport failure or Tx−timeout is again detected, then Steps 2 through 5 are repeated until the *server retries* value is exhausted or a successful response does not come from the OCS. At this point, the session is disconnected without consumption of the entire interim quota.

7. After session termination, the PCEF again sends the CCR−I in order to begin a new session. If this is successful, then the PCEF sends the CCR−T, which reports the whole temporary quota that was used.

Update−Request without Session Disconnect

Here is the message flow for this scenario:

1. The PCEF sends a CCR−U to the OCS.
2. A timeout or transport failure is detected. If transport failure is detected, then the PCEF immediately retries with the secondary server; otherwise, the Tx−expiry is triggered.
3. If the secondary server also has a transport failure or timeout, then the SU mechanism is triggered. This occurs immediately for transport failures, or after the Tx−expiry for a timeout.
4. If the interim volume and/or time are configured, then the interim quota is assigned to the session.
5. After exhaustion of the interim quota (time or volume) and if the *server retries* value is greater than zero, then the CCR−U is again sent to the server that triggered the SU mechanism. If there is another failure, a CCR−U is sent to another server that contains the entire consumed unreported quota.
6. If the transport failure or Tx−timeout is again detected, then Steps 2 through 5 are repeated until the *server retries* value is exhausted or a successful response does not come from the OCS.
7. The entire consumed quota is reported to the OCS with the successful CCR−U.
8. If the issue still exists, then the session is continued (converted to offline) or terminated as per the configuration after the exhaustion of the maximum retry value.

Update−Request with Session Disconnect

Here is the message flow for this scenario:

1. The PCEF sends a CCR−U to the OCS.
2. A timeout or transport failure is detected. If transport failure is detected, then the PCEF immediately retries with the secondary server; otherwise, the Tx−expiry is triggered.
3. If the secondary server also has a transport failure or timeout, then the SU mechanism is triggered. This occurs immediately for transport failures, or after the Tx−expiry for a timeout.
4. If the interim volume and/or time are configured, then the interim quota is assigned to the session.
5. After exhaustion of the interim quota (time or volume) and if the *server retries* value is greater than zero, then the CCR−U is again sent to the server that triggered the SU mechanism. If there is another failure, a CCR−U is sent to another server that contains the entire consumed unreported quota.
6. If the transport failure or Tx−timeout is again detected, then Steps 2 through 5 are repeated until the *server retries* value is exhausted or a successful response does not come from the OCS. At this point, the session is disconnected before it consumes the entire temporary quota.
7. The PCEF sends a CCR−T to the OCS in order to report the entire consumed quota.
8. If the OCS responds with a *2002* result code, then the additional reports are not needed.

Update−Request with Unknown Session

Here is the message flow for this scenario:

1. The PCEF sends a CCR−U to the OCS.

2. A timeout or transport failure is detected. If transport failure is detected, then the PCEF immediately retries with the secondary server; otherwise, the Tx−expiry is triggered.

3. If the secondary server also has a transport failure or timeout, then the SU mechanism is triggered. This occurs immediately for transport failures, or after the Tx−expiry for a timeout.

4. If the interim volume and/or time are configured, then the interim quota is assigned to the session.

5. After exhaustion of the interim quota (time or volume) and if the *server retries* value is greater than zero, then the CCR−U is again sent to the server that triggered the SU mechanism. If there is another failure, a CCR−U is sent to another server that contains the entire consumed unreported quota.

6. The OCS replies with a *5002* (unknown session ID) result code for the CCR−U, which is possible in the scenario where the OCS restarted and lost the session ID information.

7. The PCEF initiates a new session with the CCR−I and receives the CCA−I.

8. The PCEF reports the entire consumed interim quota via CCR−U in subsequent messages.

Update−Request with Multiple RGs (MSCC Scenario)

Here is the message flow for this scenario:

1. The PCEF sends the CCR−U for RG1 to the OCS.

2. A timeout or transport failure is detected. If transport failure is detected, then the PCEF immediately retries with the secondary server; otherwise, the Tx−expiry is triggered.

3. If the secondary server also has a transport failure or timeout, then the SU mechanism is triggered. This occurs immediately for transport failures, or after the Tx−expiry for a timeout.

4. If the interim volume and/or time are configured, then the interim quota is assigned to the session

5. At this point RG2 also exhausts the entire assigned quota but does not initiate the CCR−U because the session is already in the SU mode and begins to consume the interim quota.

6. After exhaustion of the interim quota (time or volume) and if the *server retries* value is greater than zero, then the CCR−U is again sent to the server that triggered the SU mechanism. If there is another failure, a CCR−U is sent to another server that contains the entire consumed unreported quota for both the RGs.

7. If the transport failure or Tx−timeout is again detected, then Steps 2 through 6 are repeated until the *server retries* value is exhausted or a successful response does not come from the OCS.

8. The entire consumed quota is reported to the OCS with the successful CCR−U.

9. If the issue still exists, then the session is continued (converted to offline) or terminated as per the configuration after the exhaustion of the maximum retry value.

Terminate−Request

Here is the message flow for this scenario:

1. The PCEF sends a CCR−T to the OCS.

2. A timeout or transport failure is detected. If transport failure is detected, then the PCEF immediately retries with the secondary server; otherwise, the Tx−expiry is triggered.

3. If the secondary server also has a transport failure or timeout, then the session is removed.

CCR Error Code Handling

Here is the message flow for this scenario:

1. The PCEF send a CCR to the OCS, and the OCS replies with an error code.

2. The error code is statically configured in the SU mechanism.

3. The PCEF provides the interim quota without a retry to the secondary server.

FH and SU Example Configurations

This section provides a configuration example for the FH and SU mechanisms. When both the FH and SU mechanisms are configured, the SU takes precedence over the FH for the same behavior trigger.

Here is an example:

```
credit−control group test

 diameter origin endpoint test

 diameter peer−select peer test

 quota volume−threshold percent 10

 diameter pending−timeout 80 deciseconds msg−type any

 diameter session failover

 trigger type rat lac

 apn−name−to−be−included virtual

 quota request−trigger exclude−packet−causing−trigger

 failure−handling initial−request continue retry−after−tx−expiry

 servers−unreachable initial−request terminate after−interim−volume 200
  after−interim−time 3600 server−retries 0

 servers−unreachable behavior−triggers initial−request transport−failure
  tx−expiry

 servers−unreachable update−request continue after−interim−volume 200
  after−interim−time 3600 server−retries 50

 servers−unreachable behavior−triggers update−request transport−failure
  tx−expiry
```

Verify

In order to verify that your configuration works properly, enter the ***show active−charging service <service name>*** command:

```
# show active−charging service name test


 Service name: test


  TCP Flow Idle Timeout : 300 (secs)
  UDP Flow Idle Timeout : 300 (secs)
  ICMP Flow Idle Timeout : 300 (secs)
  ICMP Flow Idle Timeout : 300 (secs)
  ALG Media Idle Timeout : 120 (secs)


 TCP Flow−Mapping Idle Timeout : 300 (secs)
 UDP Flow−Mapping Idle Timeout : Not Configured


 Deep Packet Inspection: Enabled
 Passive Mode : Disabled
 CDR Flow Control : Enabled
 CDR Flow Control Unsent Queue Size: 75
 Unsent Queue high watermark: 56
 Unsent Queue low watermark: 18


 Content Filtering: Disabled


 Dynamic Content Filtering: Disabled


 URL−Blacklisting: Disabled
 URL−Blacklisting Match−method: Exact


 Content Filtering Match−method: Generic
```

```
Interpretation of Charging-rule-base-name: active-charging-group-of-ruledefs



Selection of Charging-rule-base AVP : Last


Credit Control:

Group         : test


Mode                    : diameter

APN-name-to-be-included: gn

Trigger-Type            : N/A


Failure-Handling:

Initial-Request  : continue retry-after-tx-expiry

Update-Request   : retry-and-terminate

Terminate-Request: retry-and-terminate


Server Unreachable Failure-Handling:

Initial-Request  : terminate

Update-Request   : continue
```

Troubleshoot

Enter the ***show active-charging credit-control statistics*** command in order to view the statistics that are related to the SU and FH mechanisms. Here is a sample output:

```
#show active-charging credit-control statistics
 ...

 OCS Unreachable Stats:

 Tx-Expiry:               2291985       Response-TimeOut:        615

 Connection-Failure:      2             Action-Continue:         0

 Action-Terminated:       0             Server Retries:          2023700


 Assumed-Positive Sessions:

 Current:                 2             Cumulative:              2196851
```

Here are some important notes about this example output:

- ***Tx−Expiry*** This indicates an SU condition due to a Tx−expiry.
- ***Response−Timeout*** This indicates an SU condition due to an RT.
- ***Connection−Failure*** This indicates an SU condition due to a transport failure.
- ***Action−Continue*** This field indicates the number of sessions that went offline.
- ***Action−Terminate*** This field indicates the number of sessions that were terminated.
- ***Server Retries*** This field indicates the number of times that the OCS was retried.
- ***Assumed−Positive Sessions***:
    - ***Current*** This field indicates the number of sessions that are currently in the SU condition.
    - ***Cumulative*** This field indicates the total number of sessions that have moved into the SU status.

Enter the ***show active−charging sessions full all*** command in order to view information that is related to the SU state of the session. Here is a sample output:

```
#show active-charging sessions full all
..
..

Current Server Unreachable State:                          CCR-I

 Interim Volume in Bytes (used / allotted):                84/ 200

 Interim Time in Seconds (used / allotted):                80/ 3600

 Server Retries (attempted / configured):                   1/ 50
```

Here are some important notes about this example output:

- ***Current Server Unreachable State*** This specifies whether the current SU state is due to the CCR−I or CCR−U.
- ***Interim Volume in Bytes (used/allotted)*** This shows the interim volume in bytes used versus bytes allocated.
- ***Interim Time in Seconds (used/allotted)*** This shows the interim volume in seconds used versus seconds allocated.
- ***Server Retries (attempted/configured)*** This is the number of server retries attempted versus that configured.

Related Information

- ***Command Line Interface Reference, StarOS Release 16***
- ***Technical Support & Documentation Cisco Systems***